

## Цели

Целью данной работы является построение модели эпидемии.

## Задание

### Вариант 70

На одном острове вспыхнула эпидемия. Известно, что из всех проживающих на острове ( $N=14\ 041$ ) в момент начала эпидемии ( $t=0$ ) число заболевших людей (являющихся распространителями инфекции)  $I(0)=131$ , А число здоровых людей с иммунитетом к болезни  $R(0)=71$ . Таким образом, число людей восприимчивых к болезни, но пока здоровых, в начальный момент времени  $S(0)=N-I(0)-R(0)$ .

Постройте графики изменения числа особей в каждой из трех групп.  
Рассмотрите, как будет протекать эпидемия в случае:

1) если  $I(0) \leq I^*$

2) если  $I(0) > I^*$

## Ход работы

Опишем начальные значения согласно варианту 29 на языке Julia.

```
using Plots
using DifferentialEquations

N = 14041
I0 = 131
R0 = 71
S0 = N - I0 - R0
a = 0.01
b = 0.02
```

## Ход работы

Опишем соответствующую систему дифференциальных уравнений для первого случая, когда больные изолированы и ее решение.

```
function ode_fn(du, u, p, t)
    S, I, R = u;
    du[1] = 0
    du[2] = -b*u[2]
    du[3] = b*u[2]
end

v0 = [S0, I0, R0]
tspan = (0.0, 60.0)
prob = ODEProblem(ode_fn, v0, tspan)
sol = solve(prob, dtmax = 0.05)
S = [u[1] for u in sol.u]
I = [u[2] for u in sol.u]
R = [u[3] for u in sol.u]
T = [t for t in sol.t]
```

## Ход работы

Построим графики численности особей трех групп S, I, R.

```
plt = plot(
    dpi = 300,
    legend = :topright)

plot!(
    plt,
    T,
    S,
    label = "Восприимчивые к болезни",
    color = :red)

plot!(
    plt,
    T,
    I,
    label = "Заболевшие",
    color = :blue)

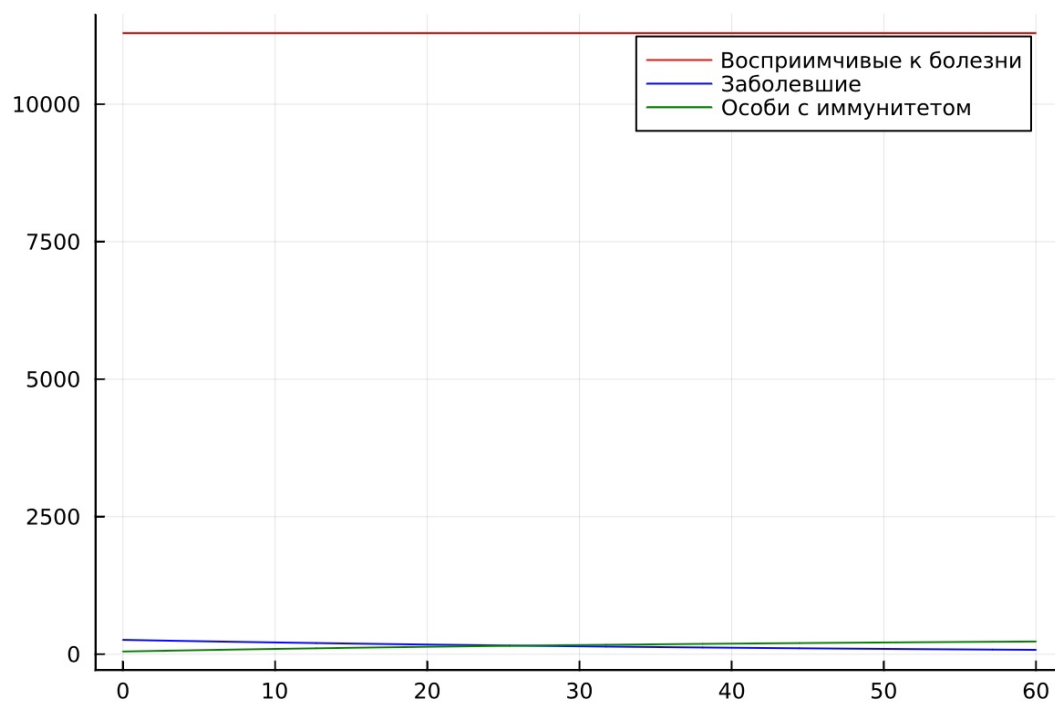
plot!(
    plt,
    T,
    R,
    label = "Особь с иммунитетом",
    color = :green)
```

## Ход работы

### Результаты работы кода на Julia

Построим графики численности особей трех групп S, I, R для первого случая (рис.1)

Out [2]:



## Ход работы

Изменим систему дифференциальных уравнений для второго случая, когда зараженные могут инфицировать особей из группы S.

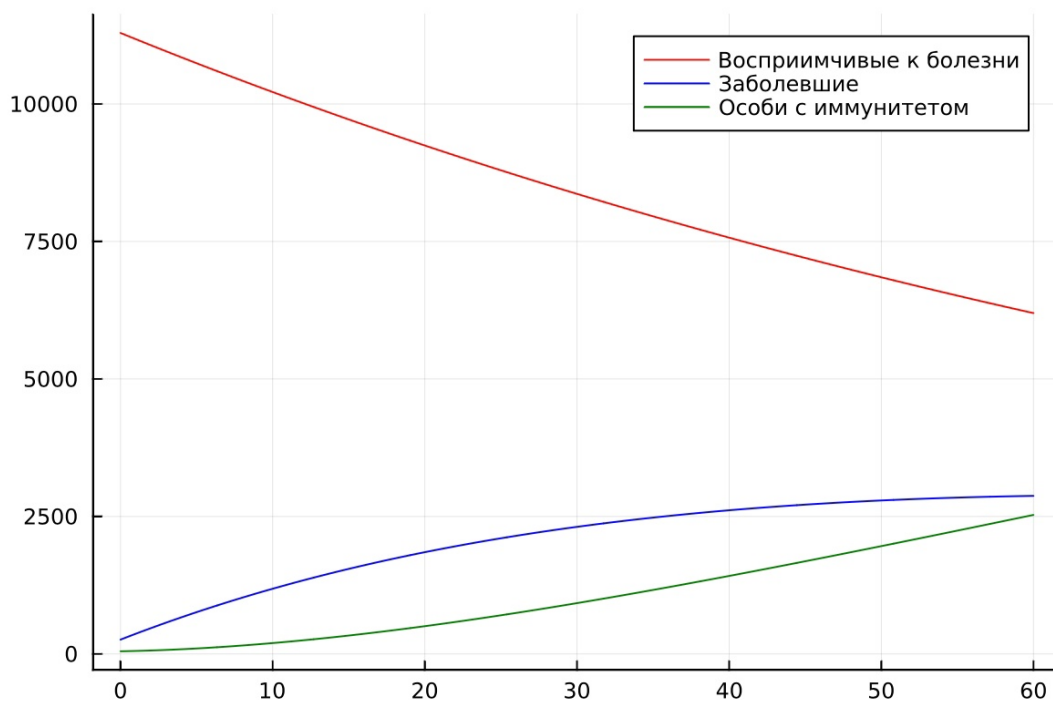
```
function ode_fn(du, u, p, t)
    S, I, R = u;
    du[1] = -a*u[1]
    du[2] = a*u[1]-b*u[2]
    du[3] = b*u[2]
end
```

## Ход работы

### Результаты работы кода на Julia

По аналогии с предыдущим построением получим графики для второго случая (рис.2)

Out [1]:



## Ход работы

Построим модель для первого случая на языке OpenModelica.

```

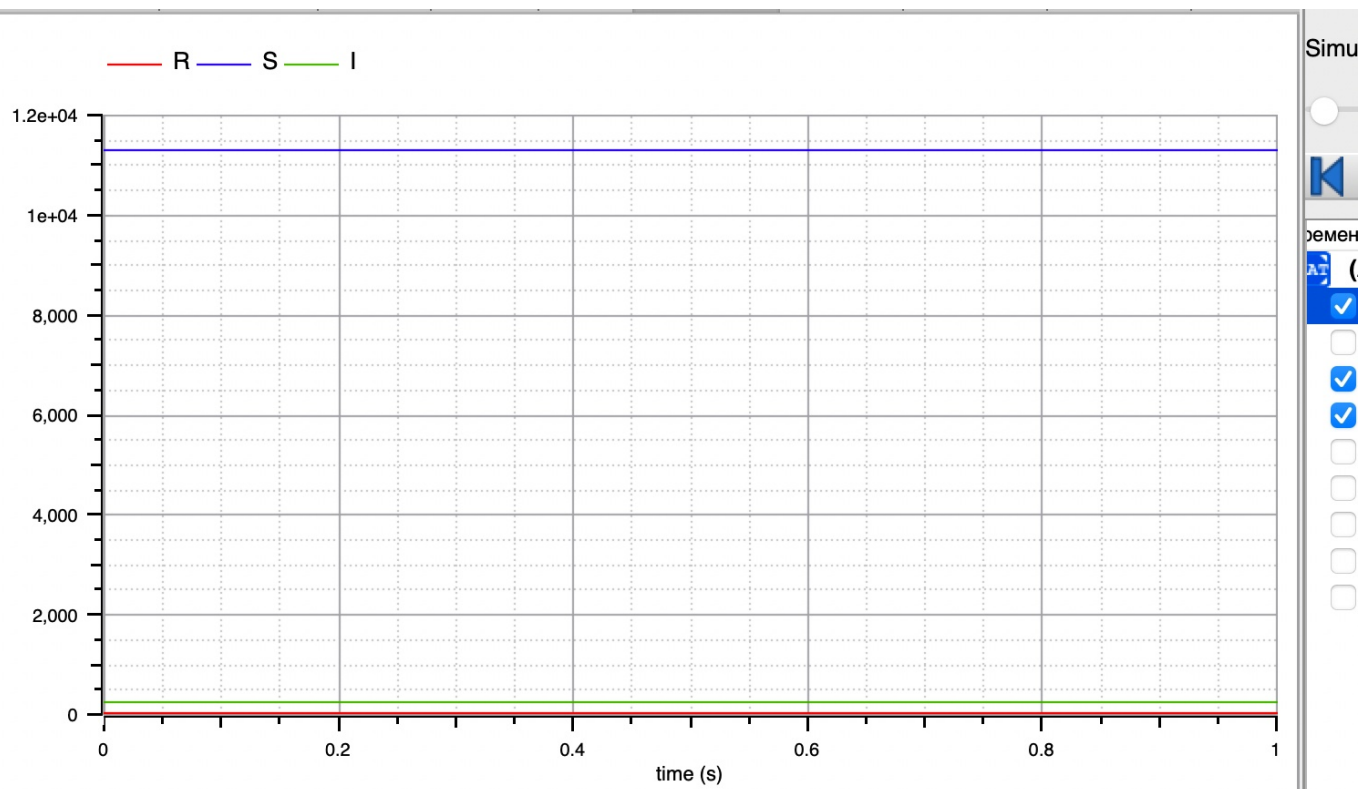
model lab6_1
  Real N = 14041;
  Real I;
  Real R;
  Real S;
  Real a = 0.01;
  Real b = 0.02;
  initial equation
    I = 131;
    R = 71;
    S = N - I - R;
  equation
    der(S) = 0;
    der(I) = -b*I;
    der(R) = b*I;
end lab6_1;

```

## Ход работы

Результаты работы кода на OpenModelica

Построим графики численности особей трех групп S, I, R для первого случая (рис.3)



## Ход работы

Для второго случая, когда зараженные могут инфицировать особей из группы S, изменим систему дифференциальных уравнений.

```

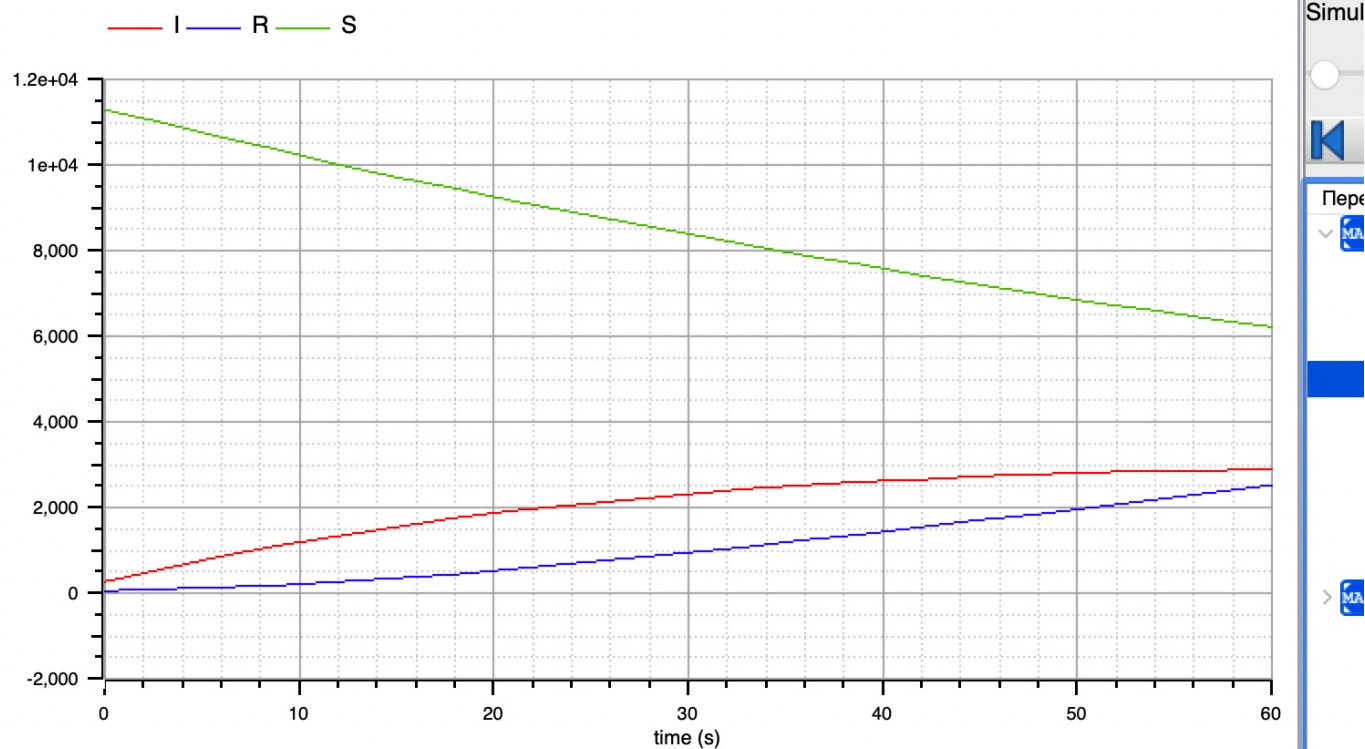
model lab6_2
Real N = 14041;
Real I;
Real R;
Real S;
Real a = 0.01;
Real b = 0.02;
initial equation
I = 131;
R = 71;
S = N - I - R;
equation
der(S) = -a*S;
der(I) = a*S-b*I;
der(R) = b*I;
end lab6_2;

```

## Ход работы

Результаты работы кода на OpenModelica

По аналогии с предыдущим построением получим графики для второго случая (рис.4)



## Результаты

В итоге проделанной работы мы построили графики зависимости численности особей трех групп S, I, R для случаев, когда больные изолированы и когда они могут заражать особей группы S, на языках Julia и OpenModelica. Построение модели эпидемии на языке OpenModelica занимает значительно меньше строк, чем аналогичное построение на Julia. Кроме того, построения на языке OpenModelica проводятся относительно значения времени  $t$  по умолчанию, что упрощает нашу работу.